**TRABAJO 2:**

**MODELOS DE PROGRAMACION LINEAL**

**ESTUDIANTE:**

**María Alejandra Marín Velásquez**

**CC 1128481980**

**DOCENTE:**

**Jaime Alejandro Ospina**

**Institución Universitaria Esumer**

**Medellín**

**2024**

****

**CONTENIDO**

* Introducción.
* Ejercicio propuesto, modelo de programación lineal.
* Solución
* Definición de variables
* Modelo matemático
* Solución Excel solver
* Interpretación
* Modelo Grafico
* Python

**INTRODUCCION**

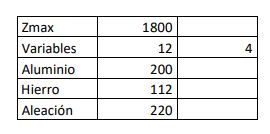
La programación lineal es una técnica de optimización que busca la solución óptima a un problema matemático, siempre y cuando esté sujeto a restricciones lineales. Esta metodología se aplica ampliamente en diversos campos como la economía, ingeniería, logística y gestión de operaciones, entre otros.

La esencia de la programación lineal es bastante sencilla: se busca la mejor solución posible dentro de un conjunto de restricciones que se expresan como ecuaciones o desigualdades lineales, lo que implica que las relaciones entre las variables involucradas en el problema son lineales en su naturaleza.

**EJERCICIO PROPUESTO**

Una fábrica de bicicletas produce dos tipos de bicicletas de carreras y de turismo, las hace en su planta de producción, usando aluminio, hierro y aleación de los dos materiales en unidades cuadradas con un mismo espesor, a saber, una bicicleta de carreras requiere 10 unidades cuadradas de aluminio, 4 de hierro y 15 de aleación, para una bicicleta de turismo, se requiere 20 unidades cuartadas de aluminio, 16 de hierro y 10 de aleación. Las bicicletas de carreras producen por su venta una ganancia de $120 y las de turismo $90. En la actualidad, la empresa dispone de 200 unidades cuadradas de aluminio, 120 de hierro y 220 de aleación. Han recibido pedidos para ambos tipos de bicicletas y les gustaría producir la cantidad de bicicletas de los dos tipos que maximice la ganancia ¿Cuántas bicicletas deben producir de cada tipo? ¿Cuánto unidades cuadradas de cada material utilizó?

**R//**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TIPO DE BICICLETAS** | **ALUMINIO** | **HIERRO** | **ALINEACION** | **GANANACIA** | **VARIABLES** |
| CARRERA | 10 | 4 | 15 | 120 | X |
| TURISMO | 20 | 16 | 10 | 90 | Y |
| DISPONIBLES | 200 | 120 | 220 |  |  |

**DEFINICION DE VARIABLES**

X= Cuántas bicicletas deben producir de carrera

Y= Cuántas bicicletas deben producir de turismo

**MODELO MATEMATICO**

* Función objetivo

Zmax=120x+90y

* Restricciones

Aluminio

10x+20y  200

Hierro

4x+16y  120

Alineación

15x+10y  220

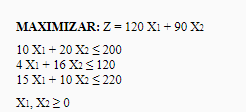
**SOLUCION EXCEL SOLVER**

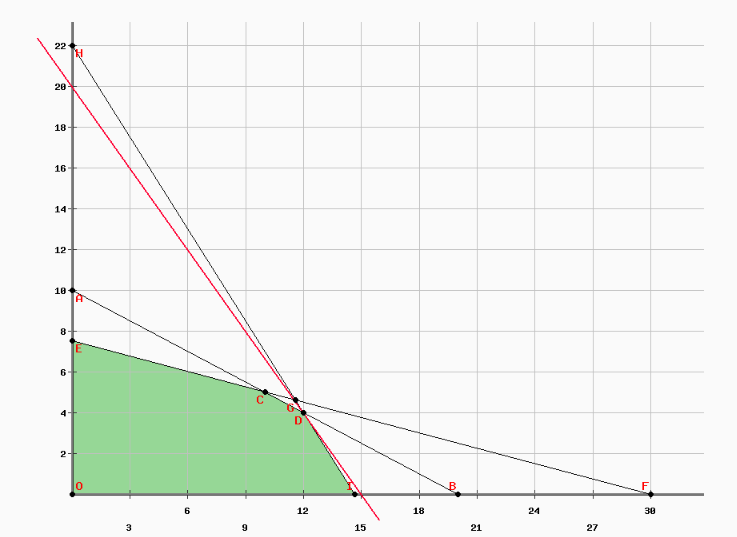
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MAX UTILIDAD** | **CARRERA** | **TURISMO** |  |  |
|  | **X** | **Y** |  |  |
| **Z MAX** | 120 | 90 | DISPONIBLE |  |
| **R1** | 10 | 20 | 200 | ALUMINIO |
| **R2** | 4 | 16 | 120 | HIERRO |
| **R3** | 15 | 10 | 220 | ALINEACION |
|  |  |  |  |  |
| **VARIABLES** | 12 | 4 |  |  |
|  |  |  |  |  |
| **Z MAX** | 1800 |  |  |  |
| **R1** | 200 | 0 |  |  |
| **R2** | 112 | 8 |  |  |
| **R3** | 220 | 0 |  |  |
|  |  |  |  |  |

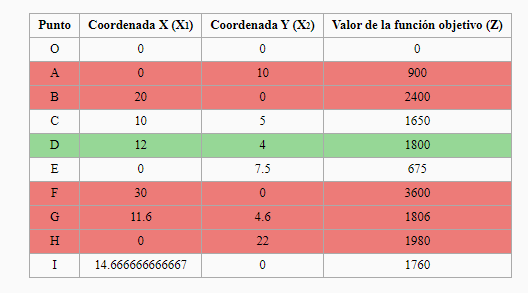
**INTERPRETACION**

En la gráfica, el punto máximo (D) representa la combinación óptima de producción que maximiza la utilidad de la fábrica de bicicletas. Esta combinación es producir 12 bicicletas de carrera (x = 12) y 4 bicicletas de turismo (y = 4). Esta producción da como resultado una utilidad máxima de $1800. Esto significa que la fábrica puede obtener la máxima ganancia produciendo esta cantidad específica de cada tipo de bicicleta. Donde la solución óptima encontrada muestra que la empresa utilizó eficientemente sus recursos disponibles (aluminio, hierro y aleación) para maximizar su utilidad. Además, al tener un sobrante de unidades de hierro, podría considerar ajustar su producción en el futuro para aprovechar mejor este recurso o incluso reducir costos si puede encontrar un uso para el excedente de hierro.

**MODELO GRAFICO**







**PYTHON**

#Modelo minimizar

import numpy as np

import matplotlib.pyplot as plt

from scipy.optimize import linprog

# Coefficients of the objective function

c = [-120, -90]

# Coefficients of the inequality constraints (left-hand side)

A = [[10, 20], [4, 16], [15, 10]]

# Right-hand side values of the inequality constraints

b = [200, 120, 220]

# Variable bounds

x\_bounds = (0, None)

y\_bounds = (0, None)

# Solve the linear programming problem

result = linprog(c, A\_ub=A, b\_ub=b, bounds=[x\_bounds, y\_bounds])

# Print the results

print("Optimal solution (number of bicycles):", result.x)

print("Optimal value of the objective function (maximum profit):", -result.fun)

# Generate x and y values for the plot

x\_values = np.linspace(0, 20, 100)

y\_values = (80 - x\_values) / 2 # from the first constraint

# Plot the constraints and solution

plt.plot(x\_values, y\_values, label="Zmax = 120x + 90y")

plt.plot(x\_values, (200 - 10 \* x\_values) / 20, label="10x + 20y <= 200 Aluminio")

plt.plot(x\_values, (120 - 4 \* x\_values) / 16, label="4x + 16y <= 120 Hierro")

plt.plot(x\_values, (220 - 15 \* x\_values) / 10, label="15x + 10y <= 220 Aleación")

plt.xlim((0, 20))

plt.ylim((0, 20))

plt.xlabel("Cantidad de bicicletas de carreras")

plt.ylabel("Cantidad de bicicletas de turismo")

plt.fill\_between(x\_values, 0, y\_values, alpha=0.1)

plt.fill\_between(x\_values, 0, (200 - 10 \* x\_values) / 20, alpha=0.1)

plt.fill\_between(x\_values, 0, (120 - 4 \* x\_values) / 16, alpha=0.1)

plt.fill\_between(x\_values, 0, (220 - 15 \* x\_values) / 10, alpha=0.1)

plt.legend()

plt.scatter(result.x[0], result.x[1], color="red", label="Solución óptima")

plt.legend()

plt.title("Solución gráfica del problema de programación lineal")

plt.show()

Gráfico, Gráfico de líneas

Descripción generada automáticamente